firewalld

NFWS 2018
Eric Garver

What is it?

- Firewall abstraction; ip{,6}tables, ebtables, ipset, and now nftables
- Default for RHEL, Fedora, SUSE, others?
- Primitives for services (e.g. ssh, mdns), ports, port forwarding, masquerading
- Simple language for custom rules, called rich rules
- Zone based
- dbus interface
- Written in python
- Commands: firewalld, firewall-cmd, firewall-offline-cmd
- GUIs: firewall-config, firewall-applet

How firewalld uses netfilter

- Calls binaries directly (iptables, ipset, nft)
- New versions use nftables and iptables simultaneously
- Assumes complete control of iptables
    - nftables backend only assumes control of “firewalld” tables

Nftables backend

- Upstream (https://github.com/firewalld/firewalld)
- Official release (v0.6.0) in the coming weeks
- Currently calls nft binary
- Requirements
  - nftables for basic features
  - Linux >=4.18 for; direct rules (NAT), AUDIT, intervals (hash:net) with timeout
- Direct interface (iptables rule passthrough) still uses iptables
- Limitations
  - Set intervals with concatenations (hash:net,port)

Recent developments

- Nftables backend
- Testsuite
    - Migrated to autotest
    - Namespaces, non-destructive to host
    - Parallel (subject to xtables lock)
    - Flake8 source code checking
- Backend abstraction
    - Don’t assume iptables throughout code
- Conversion of ipset to native nftables sets

Future development (short term)

- Migrate nftables backend to libnftables
- OUTPUT filtering via rich rules

Future development (long term)

- FORWARD filtering
- “Little snitch” like functionality, alert user to new unexpected outgoing flows
- Improved logging, NFLOG
- DOS protection (i.e. hashlimit in rich rules)
- Rich rules in services
- Code cleanups
  - Fix more flake8 reports
  - Remove fw_test.py
- Optimize rules
  - E.g. In nftables we can use “log ..” and “drop” in the same rule
  - Don’t create empty chains

What firewalld needs from netfilter

- Nftables set intervals with concatenations (e.g. ipset hast:net,port)
  - Maybe not an exact implementation, but a way to emulate would be nice
- Nftables NAT with “inet” family (low priority)
  - Currently firewalld creates tables for ip and ip6 NAT (who NATs IPv6 anyways!?)
  - Used for masquerading

How you can help

- Use it
- Report bugs! (https://github.com/firewalld/firewalld/issues)
- Fix bugs (https://github.com/firewalld/firewalld/pulls)
- Language translations (https://fedora.zanata.org/iteration/view/firewalld/master?dswid=-6716)
- New services
- Anything on “Future Development” slides
- Testsuite coverage